DEPARTMENT OF DEFENSE

DoD Enterprise Architecture Reference Model

Perspectives on a Unifying Framework for the Federal Enterprise

November 03, 2005

By

Rick Murphy, (richard.murphy@gsa.gov)
George Thomas, (george.thomas@gsa.gov)

In collaboration with the
Roy Mabry
DoD EA Congruence Community of Practice
Between April 2005 - October 2005

Perspectives on a Unifying Framework for the Federal Enterprise

Rick Murphy (richard.murphy@gsa.gov)

George Thomas (george.thomas@gsa.gov)

Overview

Common understanding and shared knowledge remain long sought after and often unrealized goals in even the highest performing organizations. Precise meaning and effective communication of even the simplest plan often requires prescriptive guidance followed by validation of the intended outcome. Interpretation of text and diagrams in architecture documents may require the skills of a semantic detective whose search for meaning turns up only vague clues of the author's intent. Effective communication across organizational boundaries whose vocabularies support missions as diverse as Federal civilian agencies and the Department of Defense (DoD) represents a more significant challenge. Despite these challenges, the rich informal semantics of natural language and free-form drawing allow today's Enterprise Architect to informally support stakeholders' alignment and investment decisions. However, consistency, integrity, and traceability as measures of Enterprise Architecture quality remain low and total cost of ownership and return on investment metrics remain elusive.

This paper builds on the GSA Enterprise Architecture team's approach to an executable Federal Enterprise Architecture (FEA) and our informal conversations with Mr. Roy Mabry (OSD) in consultation on the DoD Enterprise Architecture Business Reference Model. The sections that follow provide four perspectives on a unifying framework for the Federal Enterprise. The section on Meaning in Enterprise Architecture describes the philosophical background for specifying meaning, or precise semantics, in Enterprise Architecture. Shared Concept formalizes a common language we've heard in conversations with our colleagues. We believe this perspective from cognitive science represents the next step toward "standardizing linkages[1]" between apparently unrelated artifacts like the DoD Enterprise Architecture and the FEA reference models. Building on Shared Concept, the section on Languages, Theories, Models, and Logics provides a perspective on preserving meaning across communities using technologies from which we're building the Semantic Web. Finally, the section on Ontology Alignment, Merging, and Partitioning describes recent efforts to automate these capabilities.

Meaning in Enterprise Architecture

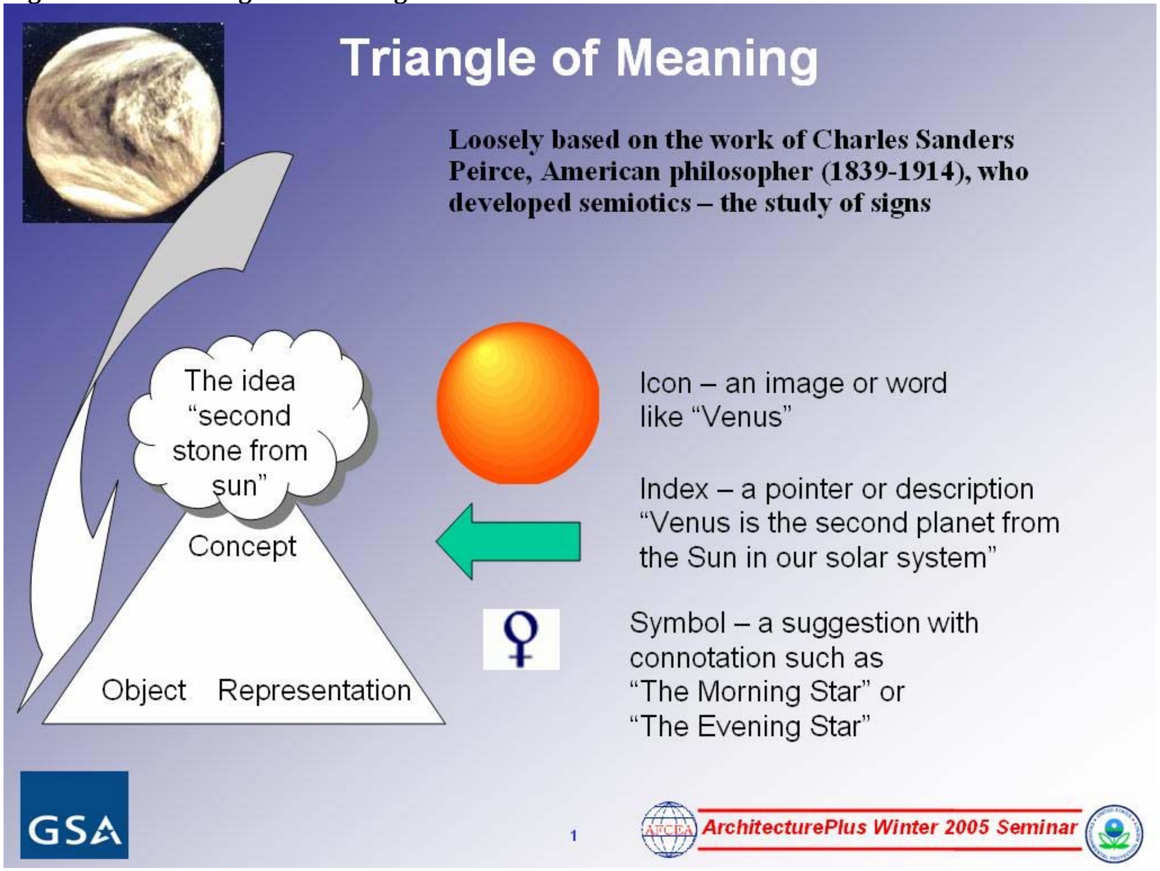
Some say even though we've invested in Enterprise Architecture we've just created another set of stovepipes. Some say we'll never standardize the linkages between the FEA reference models. We disagree. We hear a common language at conferences, in architecture review boards, in hallways, and in conversations among Enterprise Architects. We hear the words concept, term, context, and mapping every day. We've captured this common language and shown it can be formalized to support a unifying framework based on well known principles of philosophy and cognitive science. First, it's important to explain how we convey meaning, or precise semantics, in Enterprise Architecture.

Figure 1, the Triangle of Meaning, loosely based on the works of Gottlob Frege (1848-1925) and

[1] Enabling Citizen-Centered Electronic Government, 2005-2006 FEA-PMO Action Plan, Richard R. Burk, 2005

Charles Sanders Peirce (1839-1914), illustrates how concepts, the signs that represent them, and the objects themselves allow us to convey meaning. Simply put, the terms, or symbols, “The Morning Star” and “The Evening Star” had widely differing connotations when astronomers discovered they both referred to the same object. In the top-left corner is the physical object we reference: the planet Venus; the cloud at the top of the triangle describes the notion that there exists some body in the solar system, the second stone from the sun. To the right of the triangle are signs. We can represent the concept and object with three types of signs: icon, index, or symbol. We need all three object, concept, and symbol, to convey meaning in Enterprise Architecture.

Figure 1 - The Triangle of Meaning



Triangle of Meaning

Loosely based on the work of Charles Sanders Peirce, American philosopher (1839-1914), who developed semiotics – the study of signs

The idea “second stone from sun”

Concept

Object Representation

Icon – an image or word like “Venus”

Index – a pointer or description “Venus is the second planet from the Sun in our solar system”

Symbol – a suggestion with connotation such as “The Morning Star” or “The Evening Star”

Shared Concept

Based on what we know from the Triangle of Meaning, we can formalize the language used by Enterprise Architects as the Shared Concept[2] use case. This use case provides both functional and structural analyses that standardize linkages across FEA, DoD, and even NIST architectures by showing that DoD mission areas are functional areas just like FEA business areas, as are NIST security controls.

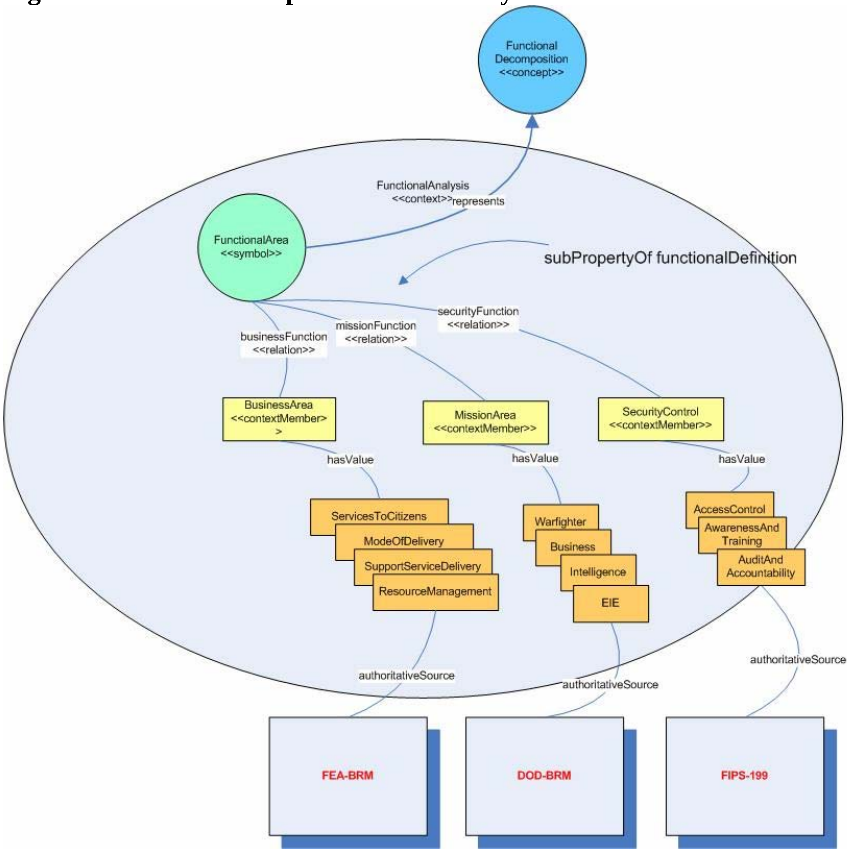
Shared Concept uses a formal notation to represent concepts, symbols, context, and mappings. In Figure 2, Shared Concept Functional Analysis, the large light-blue ellipse represents the context, dark blue circles are concepts, green circles are classes of symbols, yellow rectangles are

[2] Primer: Getting Into the Semantic Web and RDF Using N3, Tim Berners-Lee, 2000

instances, orange rectangles are values of instances, and labeled arcs are relations, or mappings. This associated meta-data is specified using double angle brackets called guilmettes.

Figure 2 illustrates how FunctionalDecomposition, a concept, is represented by FunctionalArea, a symbol. BusinessArea, MissionArea, and SecurityControl are all functional areas. Simply put, each functional area uses a different symbol to represent the concept FunctionalDecomposition. The FEA BRM provides a functional breakdown of the Federal Enterprise in four business areas: Services to Citizens, Mode of Delivery, Support Delivery of Services, and Management of Government Resources. DoD and NIST also provide functional breakdowns. DoD provides four mission areas: Warfighter, Business, Intelligence, and Enterprise Information Environment (EIE). NIST provides a functional breakdown of security as security controls, three of which are AccessControl, AwarenessAndTraining, and AuditAndAccountability. Within FunctionalAnalysis, the unifying context, the properties businessFunction, missionFunction, and securityFunction relate BusinessArea, MissionArea, and SecurityControl to FunctionalArea. To ensure traceability, the values of each functional area point to the authoritative source from which they're derived.

Figure 2 – Shared Concept Functional Analysis


Functional Decomposition <<concept>>
FunctionalAnalysis <<context>>
represents
FunctionalArea <<symbol>>
subPropertyOf functionalDefinition
securityFunction <<relation>>
missionFunction <<relation>>
businessFunction <<relation>>
BusinessArea <<contextMember>>
MissionArea <<contextMember>>
SecurityControl <<contextMember>>
hasValue
hasValue
hasValue
ServicesToCitizens
ModeOfDelivery
SupportServiceDelivery
ResourceManagement
Warfighter
Business
Intelligence
EIE
AccessControl
AwarenessAnd Training
AuditAnd Accountability
authoritativeSource
authoritativeSource
authoritativeSource
FEA-BRM
DOD-BRM
FIPS-199

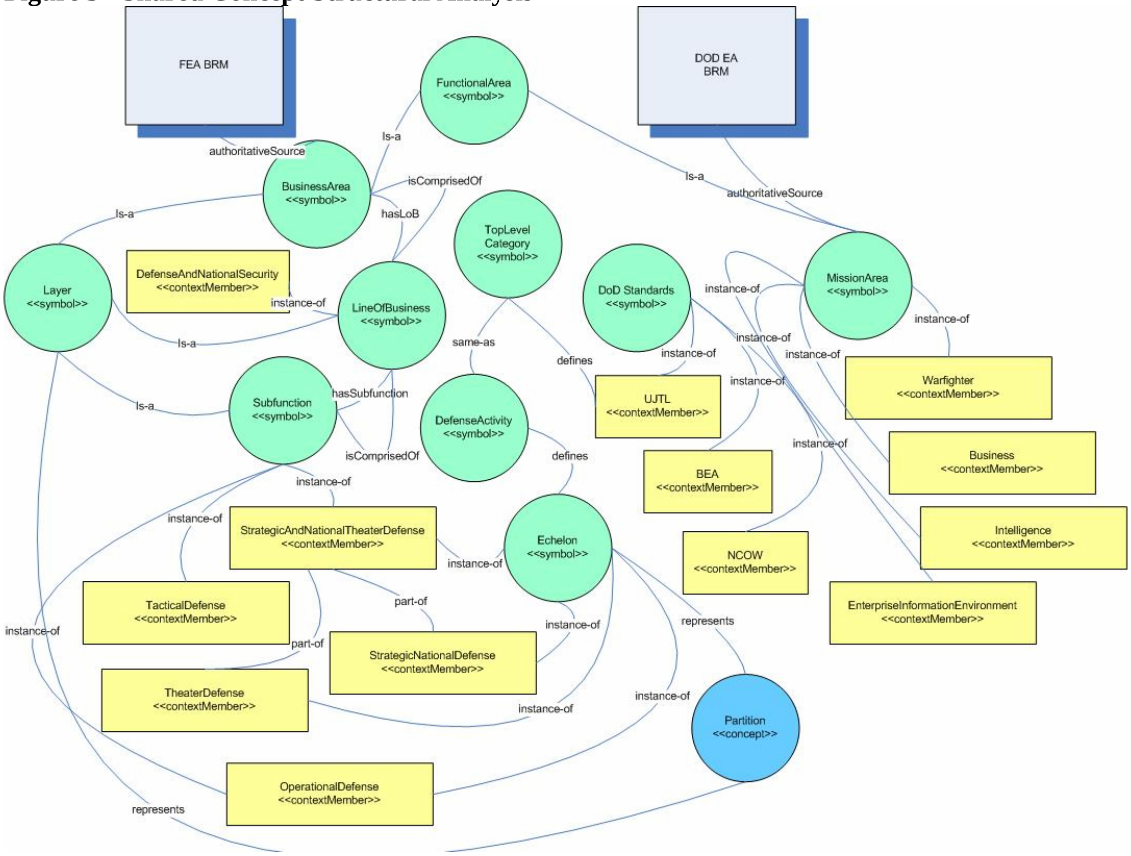
Appendix A contains two listings that demonstrate the feasibility of implementing the Shared Concept use case using executable artifacts specified in Web Ontology Language (OWL-DL) and the Pellet reasoning engine. Listing 1 is a code sample that parses, validates, and reasons over the executable artifacts. Listing 2, the output of the program, demonstrates how the reasoning engine infers the values of BusinessArea, MissionArea, and SecurityControl from functionalDefinition, a super property of businessFunction, missionFunction, and securityFunction. Properties, the labeled arcs between the nodes, are OWL language constructs that support relations and provide powerful mapping capabilities through domain and range constraints on OWL classes and individuals.

Figure 3 further illustrates the Shared Concept use case with a structural analysis of how concepts, symbols, and mappings support standardizing linkages between the FEA and DoD architectures. We use the same notation in Figure 3 as before, but we've elided the light-blue context ellipse to better layout the diagram.

The DoD Supplement to the FEA BRM and the DoD EA BRM are the authoritative sources for the Shared Concept structural analysis. We establish traceability to our functional analysis by defining BusinessArea and MissionArea with is-a relations to FunctionalArea. Is-a relations are the familiar subsumption or class inheritance relations. Following on the structure of the FEA BRM, DefenseAndNationalSecurity is an instance-of the LineOfBusiness. Instance-of means the realization of one of the class of symbols in DefenseAndNationalSecurity. The DoD Supplement to the FEA BRM defines three instances of Subfunction: StrategicNationalAndTheaterDefense, TacticalDefense, and OperationalDefense.

The DoD EA BRM describes three DoD Standards: UniformJointTaskList (UJTL), BusinessEnterpriseArchitecture (BEA), and NetCentricOperationalWarfare (NCOW). UJTL defines TopLevelCategory, which means the same as DefenseActivity. Same as means semantically equivalent. Defense activities are divided into echelons of which there are four. Strategic National Defense, Theater Defense, Tactical Defense, and Operational Defense are instances of Echelon. StrategicNationalDefense and TheaterDefense are part-of StrategicNationalAndTheaterDefense in the DoD Supplement to the FEA BRM. Echelon represents Partition, a concept reinforcing structural classification, in the DoD EA BRM as does Layer in the FEA BRM. BusinessArea, LineOfBusiness, and Subfunction are layers of the FEA BRM as specified by the is-a relation.

Figure 3 - Shared Concept Structural Analysis



Although a full treatment of the functional and structural applications of Shared Concept is beyond the scope of this paper, we have shown it's possible to specify more precise semantics than what's available in the natural language artifacts. We plan to further explore the Shared Concept use case by further specifying the functional and structural relationships between the DoD EA and its standards as executable artifacts related to the Federal Enterprise Architecture Reference Model Ontology (FEA-RMO)[3].

Languages, Theories, Models, and Logics

Today, the opportunity to unify Enterprise Architecture is greater than ever. Tomorrow, innovative methodologies backed by executable artifacts will forecast return on investment through simulation prior to acquisition. Formal traceability, or Line of Sight, through performance, business, service, technical, and data models will reduce total cost of ownership. And controlled vocabularies will increase common understanding and shared knowledge thereby reducing uncertainty in alignment and investment decisions. As the Semantic Web continues to evolve as a web of meaning; languages, theories, models, and logics become the framework for unifying our approach to Enterprise Architecture as a strategic information asset base.

The Semantic Web implies expressive formal languages such as OWL-DL which we can use to represent, validate, and reason about theories and models of knowledge in Enterprise Architecture. Logic is the science that deals with the principles and criteria of inference based on

[3] http://www.osera.gov/owl/2004/11/fea/FEA.owl

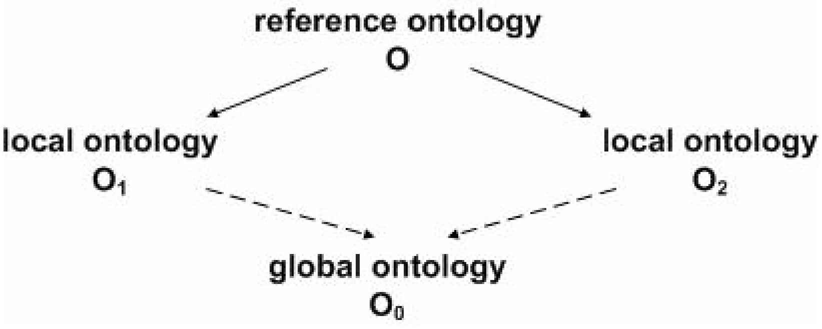
formal principles of reasoning. A logic consists of a first order language of types, together with an axiomatic system, and model-theoretic semantics. Description Logic has been shown to satisfy the practical and theoretical requirements of reasoning on the Semantic Web.[4]

The Information Flow Framework is an effort to develop the logic of distributed systems like the Semantic Web. "The Information Flow Framework celebrates the notion of a community. The standards of a community, encoded in the types and constraints of that community's ontology, represent consensual agreement *within* the community. The global standards of a collection of cooperating communities, encoded in the types and constraints of a common generic extensible ontology, represent the consensual agreement *across* communities – a standard semantics. Community ontologies, the links between such ontologies, and the resulting virtual fusion ontologies, can all be represented with the Information Flow Framework.[5]" The Information Flow Framework rejects a monolithic common upper ontology and proposes that languages, theories, models, and logics form the basis for sharing knowledge across communities.

Ontology Alignment, Merging, and Partitioning

Methods such as IF-Map[6] develop mechanisms for ontology alignment to realize information flow using local, reference, and global ontologies. Figure 4 – Ontology Mapping Scenario illustrates how to preserve meaning where two communities have encoded their knowledge in local ontologies (O_1, O_2). Each community continues to use its local ontology and has agreed on reference ontology (O_0) and mappings, represented as solid arrows, generated from the reference ontology to the local ontologies. This alignment structure (O_1 <- O_0 -> O_2) uniquely determines the global ontology (O) from which we can infer relationships that link symbols in O_1 to symbols in O_2, allowing knowledge to be shared across communities.

Figure 4 - Ontology Mapping Scenario



IF-Map describes a process to align ontologies across communities. Local ontologies in sufficiently expressive languages are first acquired, or harvested, from existing data stores. Harvested ontologies are then translated into a logic, and then mappings from the reference ontology to the local ontologies are generated by identifying regularities (Identity, Weakening, and Global Cut) in the relationships among concepts and symbols.

University of Maryland's Mindswap Group (http://www.mindswap.org) is currently developing

[4] Reducing OWL Entailment to Description Logic Satisfiability, Horrocks and Patel-Schneider, 2003.

[5] http://www.ontologos.org/IFF/IFF.html, Robert E. Kent, 2001

[6] IF-Map: An Ontology-Mapping Method based on Information Flow Theory, Kalfoglou and Schorlemmer, 2002

SWOOP, an ontology editor, and Pellet, a Description Logic reasoner for use on the Semantic Web. SWOOP and Pellet offer advanced capabilities including automated ontology partitioning and merging using ϵ-Connections. An ϵ-Connection is expressed in an Abstract Description System framework which includes Description Logic, Temporal Logic, Spatial Logic, Modal Logic, and Epistemic Logic. Given certain restrictions called ϵ-Safety, partitioning[7] is the process of refactoring modular components from an existing ontology while preserving structural and semantic compatibility. In addition to classes, properties, and their instances, a ϵ-connected ontology contains a link property which maps information in a source to a target ontology. Merging[8] is the process of creating ϵ-connected ontologies through link properties and constructing new classes.

Summary

This paper presented four perspectives from a unifying framework for the Federal Enterprise. We believe the insights we've gained from our informal discussions and investigations into the challenge of effective communication across communities as diverse as Federal civilian agencies and DoD brings us one step closer to standardizing the linkages between the DoD Enterprise Architecture and the FEA reference models. We look forward to further discussing our findings.

Appendix A

Of course our example would be incomplete without something executable. Listing 1 provides a code snippet in which we use the University of Maryland's Pellet description logic reasoner to validate the consistency of the OWL-Lite ontology and then iterate over all sub-properties of functionalDefinition. Listing 2 shows how Pellet infers each BusinessArea, MissionArea, and SecurityControl from functionalDefinition.

Listing 1 – Shared Concept Code Sample

```
Model model = ModelFactory.createOntologyModel();

// reads from url and writes to console
URL url = new URL("http://www.osera.gov/sc");
BufferedReader in = new BufferedReader(new InputStreamReader(url.openStream()));
model.read(in,null);

// creates a reasoner and binds the model to the reasoner
Reasoner reasoner = PelletReasonerFactory.theInstance().create();
InfModel inferenceModel = ModelFactory.createInfModel(reasoner,model);

// check the validity of the inference model
ValidityReport report = inferenceModel.validate();
System.out.println("model is valid: " + report.isValid() + " model is clean: " + report.isClean());

// gets the resource as an ontology class
Resource subject = model.getResource("http://www.osera.gov/sc#FunctionalDecomposition");
OntClass functionalDecomposition = (OntClass)subject.as(OntClass.class);

// list the individuals
for(ExtendedIterator i = functionalDecomposition.listInstances(); i.hasNext();){
  functionalDecompositionIndividual = (Individual)i.next();
  System.out.println("functional area individual is: " + PrintUtil.print(functionalDecompositionIndividual));
}
```

[7] Automatic Partitioning of OWL Ontologies Using ϵ-Connections, Grau, Parisa, Sirin, Kalyanpur, 2005

[8] Combining OWL Ontologies Using ϵ Connections, Grau, Parisa, Sirin, 2005

```
                    // get the individuals of the class that satisfy the functionalDescription property
                    for(StmtIterator i =
functionalDecompositionIndividual.listProperties(model.getProperty("http://www.osera.gov/sc#functionalDefinition")); i.hasNext();){
                      Statement statement = (Statement)i.next();
                      System.out.println("property statement is: " + PrintUtil.print(statement));
                    }
```

Listing 2 – Shared Concept Stack Trace

```
   [java] model is valid: true model is clean: true
   [java] functional area individual is: http:// localhost /sc#FunctionalArea
   [java] property statement is: (http://localhost/sc#FunctionalArea http://localhost/sc#functionalDefinition
http://localhost/brm#ServicesToCitizens)
   [java] property statement is: (http://localhost/sc#FunctionalArea http://localhost/sc#functionalDefinition
http://localhost/brm#ModeOfDelivery)
   [java] property statement is: (http://localhost/sc#FunctionalArea http://localhost/sc#functionalDefinition http://localhost/fips-
199#AccessControl)
   [java] property statement is: (http://localhost/sc#FunctionalArea http://localhost/sc#functionalDefinition http://localhost/fips-
199#AwarenessAndTraining)
   [java] property statement is: (http://localhost/sc#FunctionalArea http://localhost/sc#functionalDefinition http://localhost/fips-
199#AuditAndAccountability)
   [java] property statement is: (http://localhost/sc#FunctionalArea http://localhost/sc#functionalDefinition
http://localhost/osd#Intelligence)
   [java] property statement is: (http://localhost/sc#FunctionalArea http://localhost/sc#functionalDefinition
http://localhost/osd#Warfighter)
   [java] property statement is: (http://localhost/sc#FunctionalArea http://localhost/sc#functionalDefinition
http://localhost/osd#Business)
   [java] property statement is: (http://localhost/sc#FunctionalArea http://localhost/sc#functionalDefinition
http://localhost/osd#EnterpriseInformationEnvironment)

BUILD SUCCESSFUL
Total time: 23 seconds
[rmurphy@localhost ~/sc]$
```